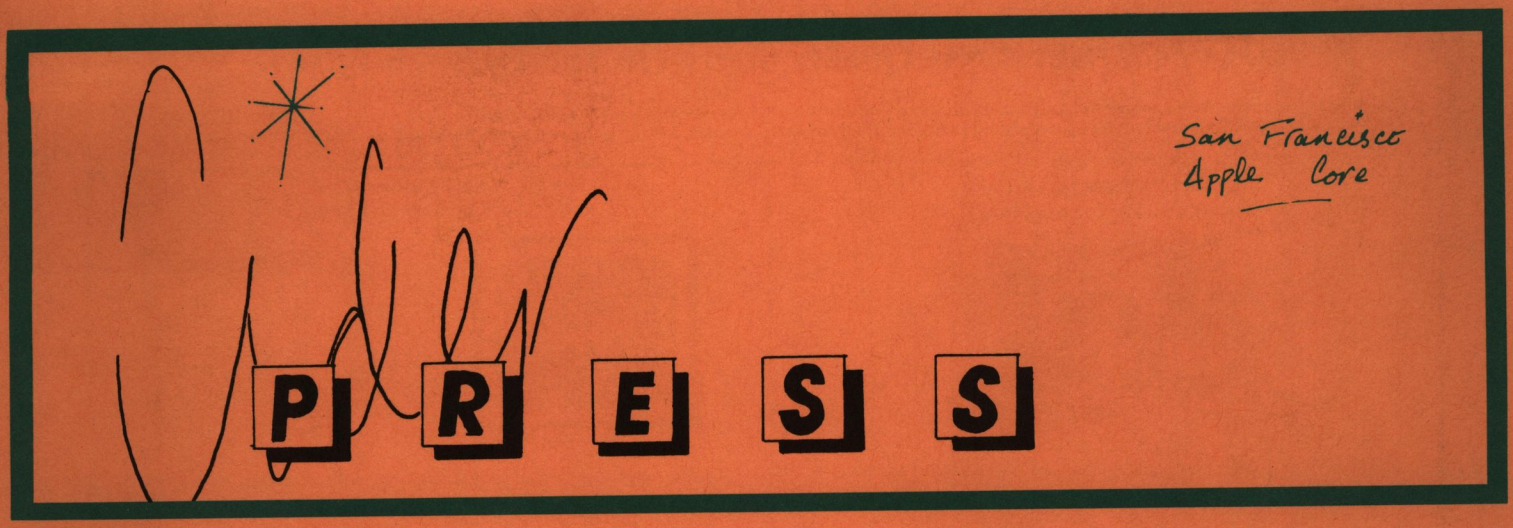


San Francisco Apple Core

IN THIS ISSUE

REM 1
B.A.S.I.C. - Joyce Conklin 2
Pascal Pages
Three Easy Execs - Max Nareff 6
Abbreviated Keys for Random Access - Max Nareff . . . 7
Review
The Graphic Solution - Jerry Matsukado 9
Poke of the Month - Kathy Rudovsky 11
Interest Group News 12
SWAP MEET 13
Thanks & Annual Report 14
Interest Group Leaders 15

July, 1984

GENERAL MEETINGS:

August 18

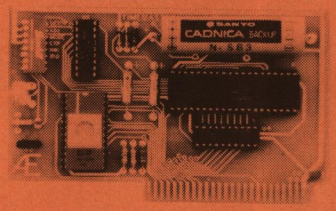
Swap Meet! - September 15

APPLIED ENGINEERING Quality Starts On The Drawing Board....

But it doesn't end there. Our commitment to excellence applies all the way down the line, through manufacturing and testing to our three year no-hassle warranty. The aim, of course, is to provide you with the best designed and best crafted hardware available. Try any of the peripherals below and see what it's like to own the finest.

The TIMEMASTER
Finally, a clock that does it ALL!

Automatically date stamps files with PRO-DOS



Fully PRO-DOS compatible

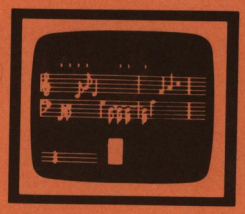
- Just plug it in and your programs can read the year, month, date, day, and time to 1 millisecond! The only clock with both year and ms.
- A rechargeable NiCad battery will keep the TIMEMASTER running for over ten years.
- Powerful 2K ROM driver — No clock could be easier to use.
- Full emulation of most other clocks, including Mountain Hardware's Appleclock (but you'll like the TIMEMASTER mode better).
- Basic, Machine Code, CP/M and Pascal software on 2 disks!
- Eight software controlled interrupts so you can execute two programs at the same time (many examples are included).
- On-board timer lets you time any interval up to 48 days long down to the nearest millisecond.

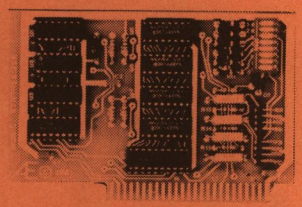
The TIMEMASTER includes 2 disks with some really fantastic time oriented programs (over 25) plus DOS dater so it will automatically add the date when disk files are created or modified. The disk is over a $200.00 value alone — we give the software others sell. All software packages for business, data base management and communications are made to read the TIMEMASTER.

If you want the most powerful and the easiest to use clock for your Apple, you want a TIMEMASTER.

PRICE $129.00

Super Music Synthesizer
Improved Hardware and Software

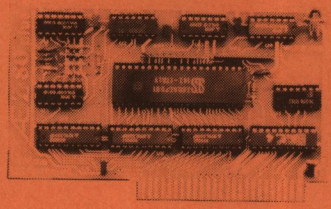




- Complete 16 voice music synthesizer on one card. Just plug it into your Apple, connect the audio cable (supplied) to your stereo, boot the disk supplied and you are ready to input and play songs.
- It's easy to program music with our compose software. You will start right away at inputting your favorite songs. The Hi-Res screen shows what you have entered in standard sheet music format.
- Now with new improved software for the easiest and the fastest music input system available anywhere.
- We give you lots of software. In addition to Compose and Play programs, 2 disks are filled with over 30 songs ready to play.
- Easy to program in Basic to generate complex sound effects. Now your games can have explosions, phaser zaps, train whistles, death cries. You name it, this card can do it.
- Four white noise generators which are great for sound effects.
- Plays music in true stereo as well as true discrete quadraphonic.
- Full control of attack, volume, decay, sustain and release.
- Will play songs written for ALF synthesizer (ALF software will not take advantage of all our card's features. Their software sounds the same in our synthesizer.)
- Our card will play notes from 30HZ to beyond human hearing.
- Automatic shutoff on power-up or if reset is pushed.
- Many many more features.

PRICE $159.00

Z-80 PLUS!



- TOTALLY compatible with ALL CP/M software.
- The only Z-80 card with a special 2K "CP/M detector" chip.
- Fully compatible with microsoft disks (no pre-boot required).
- Specifically designed for high speed operation in the Apple IIe (runs just as fast in the II+ and Franklin).
- Red "CP/M WORKING" LED indicator, the Z-80 Plus does not interfere with non-CP/M programs.
- A semi-custom I.C. and a low parts count allows the Z-80 Plus to fly thru CP/M programs at a very low power level. (We use the Z-80A at fast 4MHZ.)
- Does EVERYTHING the other Z-80 boards do, plus Z-80 interrupts.

Don't confuse the Z-80 Plus with crude copies of the microsoft card. The Z-80 Plus employs a much more sophisticated and reliable design. With the Z-80 Plus you can access the largest body of software in existence. Two computers in one and the advantages of both, all at an unbelievably low price.

PRICE $139.00

Viewmaster 80

There used to be about a dozen 80 column cards for the Apple, now there's only ONE.

- TOTALLY Videx Compatible.
- 80 characters by 24 lines, with a sharp 7x9 dot matrix.
- On-board 40/80 soft video switch with manual 40 column override
- Fully compatible with ALL Apple languages and software—there are NO exceptions.
- Low power consumption through the use of CMOS devices.
- All connections on the card are made with standard video connectors.
- All new design (using a new Microprocessor based C.R.T. controller) for a beautiful razor sharp display.
- The VIEWMASTER incorporates all the features of all other 80 column cards, plus many new improvements.

	PRICE	BUILT IN SOFTSWITCH	SHIFT KEY SUPPORT	LOW POWER DESIGN	THREE YEAR WARRANTY	7x9 DOT MATRIX	LIGHT PEN INPUTS	40 COLUMN OVERRIDE	INVERSE CHARACTERS
VIEWMASTER	169	YES	YES	YES	YES	YES	YES	YES	YES
SUP'RTERM	MORE	NO	YES	NO	NO	NO	NO	YES	YES
WIZARD80	MORE	NO	NO	NO	NO	YES	NO	YES	YES
VISION80	MORE	YES	YES	NO	NO	YES	NO	NO	NO
OMNIVISION	MORE	NO	YES	NO	NO	NO	NO	YES	YES
VIEWMAX80	MORE	YES	YES	NO	NO	YES	NO	NO	YES
SMARTERM	MORE	YES	YES	NO	NO	NO	YES	YES	NO
VIDEOTERM	MORE	NO	NO	YES	NO	YES	YES	NO	YES

The VIEWMASTER 80 works with all 80 column applications including CP/M, Pascal, WordStar, Format II, Easywriter, Apple Writer II, VisiCalc, and all others. The VIEWMASTER 80 is THE MOST compatible 80 column card you can buy at ANY price!

PRICE $169.00

MemoryMaster IIe 128K RAM Card

- Expands your Apple IIe to 192K memory.
- Provides an 80 column text display.
- Compatible with all Apple IIe 80 column and extended 80 column card software (same physical size as Apple's 64K card).
- Can be used as a solid state disk drive to make your programs run up to 20 times FASTER (the 64K configuration will act as half a drive).
- Permits your IIe to use the new double high resolution graphics.
- Automatically expands Visicalc to 95 K storage in 80 columns! The 64K config. is all that's needed, 128K can take you even higher.
- PRO-DOS will use the MemoryMaster IIe as a high speed disk drive.
- Lowest power consuming 128K card AVAILABLE.
- Complete documentation included, we show you how to use all 128K.

If you already have Apple's 64K card, just order the MEMORYMASTER IIe with 64K and use the 64K from your old board to give you a full 128K. (The board is fully socketed so you simply plug in more chips.)

MemoryMaster IIe with 128K $249
Upgradeable MemoryMaster IIe with 64K $169
Non-Upgradeable MemoryMaster IIe with 64K $149

Our boards are far superior to most of the consumer electronics made today. All I.C.'s are in high quality sockets with mil-spec. components used throughout. P.C. boards are glass-epoxy with gold contacts. Made in America to be the best in the world. All products work in the APPLE IIE, II, II+ and Franklin. The MemoryMaster IIe is IIe only. Applied Engineering also manufactures a full line of data acquisition and control products for the Apple; A/D converters and digital I/O cards, etc. Please call for more information. All our products are fully tested with complete documentation and available for immediate delivery. All products are guaranteed with a no hassle **THREE YEAR WARRANTY.**

All Orders Shipped Same Day
Texas Residents Add 5% Sales Tax
Add $10.00 If Outside U.S.A.
Dealer Inquiries Welcome

Send Check or Money Order to:
APPLIED ENGINEERING
P.O. Box 470301
Dallas, TX 75247

Call (214) 492-2027
7a.m. to 11p.m. 7 days a week
MasterCard, Visa & C.O.D. Welcome
No extra charge for credit cards

CIDER PRESS

REM *Timeliness* and *Lateness* are ingredients of the Human Condition (q.v.). These concepts apply, in their most pure sense, only to the activities of Man's own creation. If it is still cold and raining in March or April, we say *"Spring is late!"* It is the arrival of Spring which sets the start of SpringTime in the natural order of things, not some external schedule. The Sun *"rises"* in the morning - but does it? Rather, it is morning when the sun rises! The regularity of these events guides our lives and we depend on it.

It is fitting that we should rely on forces greater that ourselves to set the cadence and rhythm of our lives. That too is the natural order of things. All other things are secondary and subject to the flaws which humans so exquisitely posess. The energy and force of human endeavour are not so dependable nor as constant as the rhythm of the tides or the oscillations of the Cesium atom, but much more varied. That variety, in flavour, intensity and timing is part of the richness that we add to the framework we percieve.

If the *Cider Press* should not arrive on schedule, remember, it is not the Sun, nor the seasons - but rather a collection of human energies and volunteer efforts. We try. We are human.

THE CORE'S CORE

Directors

Sami Capron	415/388-8631
Joyce Conklin	415/341-6955
Michael Fischer	415/933-4435
Alan Morris	415/566-2342
Kathy Rudovsky	415/756-6480
Don Scellato	415/672-2051
Alan L Smerican	415/566-2342
Allan B. Woods	415/687-9659

Service/Committees

Membership	Alan L Smerican Ted Chatterton	415/566-2342
Cider Press -		
Editor	Allan Woods	415/687-9659
BASIC	Joyce Conklin	415/341-6955
Marketing Dir.	Don Scellato	415/672-2052
General Meeting	Michael Fischer	415/933-4435
Treasurer	Alan Morris	415/566-2342
Interest Groups	Kathy Rudovsky	415/756-6480

CIDER PRESS POLICY

All manuscripts, photographs, and other materials will be considered submitted free and released for publication. They become the property of the San Francisco Apple Core and the CIDER PRESS. Authors should clearly mark all material submitted for publication so that proper credit can be given.

The views expressed in any articles are those of the authors, and not necessarily those of the San Francisco Apple Core, Inc., the Editor, Club Officers or Directors . . . or anyone else in their right minds.

The San Francisco Apple Core Incorporated is a non - profit organization comprised of and supported by owners and users of Apple I I and Apple / / / computers. The Club is run by officers and committees, and the Club endeavors to aid Apple owners through its Program Libraries, Program Disks, meetings, and this publication.

Article Submission: The CIDER PRESS welcomes any and all articles dealing with Apple computers and their associated hardware and software. Articles may be submitted (typed or printed, double - space), or as disk text files using any of the popular word processors or text editors. Any author who has an article published in the CIDER PRESS will receive a free copy of the most current Disk - of - the - Month.

Reprints: All articles appearing in the CIDER PRESS not copyrighted by the author may be reprinted by other *non - profit* Apple User Groups, without prior permission, so long as proper credit is given to both the author and the CIDER PRESS. Permission to reprint a copyrighted article must be obtained in advance, by writing to the authors, c/o the San Francisco Apple Core.

San Francisco Apple Core
1515 Sloat Blvd.
San Francisco, CA 94132

B*A*S*I*C

Copyright (c) 1984, by Joyce Conklin

Since it is now summertime (or at least it will be by the time you read this!) I thought it would be interesting to make the column a few Applesoft listing which are instructive on several odd points.

I know that the editor doesn't particularly like page after page of program listings, but once in a while, I think you learn more by trying a couple of programs to see what they do than just reading about programming!

In the Jan. 1983 issue of the Cider Press I addressed the issue of PEEKs and POKEs - what they were and how they worked. This first listing shows how to drive your friend nuts by turning on the disk drive and then turning it off, even tho' it's not doing anything execpt spinning!

Hide a few of these lines in a program which you share and watch the consternation. Incidentally, all these listings came from a friend who only knew he got them from yet another friend - they do not have any copyright statements or even any REM statements to show the source.

As they are all in open Applesoft, they aren't capable of being protected, but I would hate to think that they were at one time copyright material! May I make a plea with each of you when you type in programs from magazines which you have purchased that you include at least a one or two line REM to indicate the source and any possible copyright.

Obviously, when you purchase a magazine, all the programs are yours to type in, and in spite of the pleas of Mike Harvey (editor of Nibble), I doubt that anyone is going to prevent you from sharing your magazine, or even sharing the programs which you typed. However, it is nice to know who was the original author.

I remember reading a catalog program in a magazine one month which also included a statement of the FREe space available each time you CATALOGed the disk. Within a very few weeks, this same program, even to the same line numbers, showed up as the CATALOG program on the SFAC disk of the month!! Without the credit, too.

In the literary world this is called plagerism; there are more swashbuckling names for it in the software world!

Enough of the lecturing .... on with the programs. Comments which would ordinarily appear in REM statements will show in lower case.

```
20  SPEED= 255
    Normal display speed

30  TEXT : CALL  - 936
    The same as HOME; people who used to
    program in Integer BASIC used this
    command because HOME was not avail-
    able until the implementation of
    Applesoft

40  VTAB 2: PRINT "DRIVE ON/OFF"

50  VTAB 6: PRINT "POKE 49385,0 - TURNS
    DRIVE ON"

60  SPEED= 70: VTAB 8: PRINT "JPOKE
    49385,0"
    This slows down the speed with which
    characters are displayed on the
    screen

70  POKE 49385,0

80  FOR X = 1 TO 2000: NEXT X
    Delay loop or empty loop

90  SPEED= 255: VTAB 11: PRINT "POKE
    49384,0 - TURNS DRIVE OFF"

100  SPEED= 70: VTAB 13: PRINT "JPOKE
     49384,0"

110  POKE 49384,0

120  FOR X = 1 TO 2000: NEXT X

130  VTAB 24
     Moves cursor to the bottom of the
     screen
```

```
140  SPEED= 255

150  END
```

I can't think of a useful function for this POKE, but it will sure drive your friends crazy if they aren't in on the game!

On the other hand, the following three listings are all ways of disabling the RESET key, so that if someone tries to stop your program and list it by first breaking in with <RESET>, they won't have any luck.

Each of the listings appears as a program, so if you want to type the whole thing in and save it, you can then run the program and the instructions will appear on the screen.

```
20  TEXT : HOME

30  HTAB 9: PRINT "5 REM  RESET=RUN"

40  HTAB 9: PRINT "10 POKE 1010,102"

50  HTAB 9: PRINT "20 POKE 1011,213"
```

1010 and 1011 are the memory locations for the Autostart ROM reset "vector"

```
60  HTAB 9: PRINT "30 CALL -1169"
```

Location -1169 ($FB6F or 64367) is the beginning of the ROM routine to "set power condition" for Autostart

```
70  REM  CALL -1169

80  HTAB 9: PRINT "40 REM IF <<ERR>>"

90  HTAB 9: PRINT "50 ONERR GOTO 10"

100  POKE 34,6
```

Protects the first six lines of the screen display from scrolling off

------ ------ -------

```
20  TEXT : HOME

30  HTAB 9: PRINT "5 REM IF <<ERR>>"

40  HTAB 9: PRINT "10 ONERR GOTO 20"

50  HTAB 9: PRINT "15 REM RESET=RUN"

60  HTAB 9: PRINT "20 POKE 1110,102"

70  HTAB 9: PRINT "30 POKE 1111,213"

80  HTAB 9: PRINT "40 POKE 1112,112"
```

In the manual "What's Where in the Apple, these locations show as part of the page 1 or text screen display.

But doing these POKEs and then pressing Control-Reset reboots the //e, sure enough - it doesn't take Control-Open Apple-Reset.

```
90  POKE 34,7
```

----- ----- -----

```
20  TEXT : HOME

30  VTAB 2: HTAB 15: PRINT "RESET III"

40   VTAB  6:  HTAB  12:  PRINT  "10  POKE
40286,35"


50  HTAB 12: PRINT "20 POKE 40287,216"
```

Locations 40286-87 are "Current Language entry vector to 'soft entry'"

```
60  HTAB 12: PRINT "30 ONERR GOTO A"

70   VTAB 12: HTAB 2: PRINT "NOW W";"HEN YOU
PRESS <CTRL-C> OR <R";"ESET>"

80   HTAB  2:  PRINT "IT GOES TO WHA";"TEVER
'A' IN LINE 30 IS"

90  FLASH : VTAB 8: HTAB 28: PRINT "<-"

100  FOR X = 1 TO 2000: NEXT X

110  NORMAL
```

Stops the flashing display

```
120  VTAB 8: HTAB 28: PRINT "  "
```

Deletes the flashing arrow

```
130  VTAB 23
```

Moves the cursor to the bottom of the screen

```
140  END
```

This last routine allows you to re-direct the user to wherever in the program you wish, rather than just starting the program over again.

----- ----- -----

This next program will allow you to catalog your disk and the control characters which may have been hidden within the catalog names will now show up in inverse characters. A very handy utility when you forget which control characters you hid! Or when they got in there by error.

```
20  DATA   201,141,240,21,201,136

30  DATA  240,17,201,128,144,13

40  DATA  201,160,176,9,72,132
```

```
50  DATA  53,56,233,64,76,249

60  DATA  253,76,240,253

70  FOR I = 768 TO 768 + 27

80  READ V: POKE I,V: NEXT I
```

Loads the 28 data items into the consecutive address locations beginning at 768

```
90  POKE 54,0: POKE 55,3
```

54 and 55 are the address of the ROM routine which receives and disposes of output characters

```
100  CALL 1002
```

Reset DOS hooks i.e saves addresses of character input and output routines currently in use and reconnects DOS

```
110  CALL 42350
```

Location of the ROM routine to run "CATALOG" command

Location 768 is a very useful address to remember; it is also the place where you store the sound generating routine for the Apple (maybe I will include that in a later column -- until then, if you are interested, look it up in "The Apple User's Encyclopedia" by Gary Phillips, JOYCE CONKLIN, and DON SCELLATO -- plug, plug!!!)

After doing a little bit of programming, you may dream of writing the greatest integrated software package ever and retiring on your earnings. Unless you have lots of faith and guts, like Mark Pelzarski of Penguin Software who publishes all his applications programs unprotected, there will come a day when you want to protect your program from any copying.

Probably a futile attempt, because every new protection scheme fosters a new protection breaker! But you can slow people down a bit by the following:

A. Type POKE 44033,16 (in the immediate mode) and press <Return>
B. Insert a new disk in your drive and INIT it

What you have done is to change the track of your catalog, INCLUDING the VTOC (Volume Table of Contents, which is how the DOS knows where to go to find the rest of the Catalog), from track 17 (the normal location) to track 16. Other numbers from 3 through 15 inclusive will also work, moving the catalog to the track corresponding to the number which you POKEd into 44033.

To transfer programs from a normal (track 17 catalog) disk to the new disk, do the following:

1. Boot a normal disk
2. Load a program from the normal disk
3. Type "POKE 44033,16" (assuming you used 16 in the INIT process) and press <Return>
4. Swap for your protected disk (or have that one in your second drive)
5. SAVE your program to the protected disk
6. Type "POKE 44033,17" to return things to normal
7. Insert your normal disk (if you are using a single-drive system) and repeat from step 2 as needed.

Tricky, huh?

The last listing is an interesting HI-RES graphics display, showing a bar-graph in three dimensions. The bars appear from right to left. Can anyone out there figure out how to have the bars appear from left to right, as you might want them for a business graph or some such?

```
20  HGR2 : HCOLOR= 1

30   HPLOT  1,0 TO 1,139 TO 253,139 TO 253,0
TO 1,0
```

Draws a border on the screen

```
40   HPLOT 25,178 TO 279,178: HPLOT 0,139 TO
25,165 TO 25,191
```

Draws a part of the "step" below the border

```
50   HPLOT  25,165  TO  279,165  TO 253,139:
HPLOT 279,165 TO 279,191 TO 25,191
```

Completes the step below the border

```
60  HCOLOR= 3: FOR Y = 10 TO 130 STEP 10

70  HPLOT 5,Y: NEXT
```

Plots tick marks along the left edge of the border; originally the color in the program was '2'; I had to change it to '3' to get the dots to show

```
80  FOR X = 255 TO 30 STEP  - 14

90  S  =  INT ( RND (1) * 140 + 3):R = 161 -
S

100  FOR Y = R TO 160

110   HCOLOR=  2: HPLOT X - 15,Y - 15 TO X -
1,Y
```

Plots diagonal line of top of first bar on the right

```
120  HCOLOR= 3: HPLOT X,Y TO X + 10,Y
```

Plots the horizontal line at the top of the first bar

```
130  NEXT : HCOLOR= 1
```

Completes the first bar

```
140  FOR Y = R - 1 TO R - 15 STEP  - 1

150   HPLOT X - (R - Y) + 1,Y TO X - (R - Y)
+ 11,Y: NEXT : NEXT
```

The first 'NEXT' in line 150 completes the loop which draws the top of the bar, making it look 3-D The second 'NEXT' draws the rest of the bars. It goes with the 'FOR X' in line 80

```
160  END
```

----- ----- -----

How about this "problem" with the 80-column card in the Apple //e? RUN the short program below and then make the modifications shown and RUN it again. Have you any explanation for what happens? Why can't you tab past 40 in the same line but it appears to do so when each statement is on a separate line?

```
20  PR# 3
30  FOR L = 1 TO 80
40  PRINT  INT (L / 10);
50  NEXT
60  FOR L = 1 TO 8
70   FOR  M = 1 TO 9: PRINT M;: NEXT : PRINT
"0";
80  NEXT
90  PRINT  TAB( 4)"4"
100  PRINT  TAB( 24)"24"
110  PRINT  TAB( 39)"39"
120  PRINT  TAB( 47)"47"
130  PRINT  TAB( 78)"78
```

The only modification is to put a semicolon at the end of each of the program lines 90 through 120. Notice that the spacing of even the first three is as if they were being printed in 40-column display.

Wha' happened? Oh yeah - the PR#3 also seems to disable the ordinary DOS hooks or some such because PLE will not work in the 80 column mode but is there waiting for you after you press Control-Reset.

If you work with these items for the month of July you will probably be further along than I will be in my projects! It seems that I have a lot of good intentions but not much organization. Have a good time with your Apple and take advantage of the sun.THATS ALL FOLKS....

LOW PRICE DISK DRIVES !!!!

DISTAR 5-1/4 INCH HALF HIGH DRIVES

DIRECT DRIVE MOTOR.....

AUTOMATIC DOS SELECTION.....

POSITIVE LOCK LEVER.....

6 MONTH WARRANTY.....

$160 + TAX

PURCHASE AT MEETING OR ORDER FROM THE SFAC BY MAIL.....

LOW PRICE DISK DRIVES !!!!

AMT 5-1/4 INCH HALF HIGH DRIVES

DIRECT DRIVE MOTOR.....

PANASONIC MECHANISM.....

WARRANTY.....

$189 + TAX

PURCHASE AT MEETING OR ORDER FROM THE SFAC BY MAIL.....

THREE EASY EXECS
By
Max J. Nareff

Exec files are as easy to prepare in Pascal as they are in BASIC. When incorporated in a program they expand its range and capabilities. A few simple Exec files are presented.

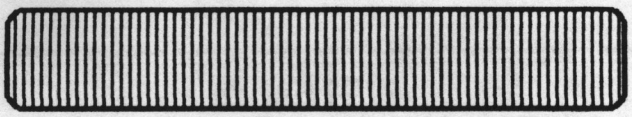
1. Date: This short program activates the F(iler, and the D(ate prompt. After compilation, the code segment may be C(hanged to SYSTEM.STARTUP so that the D(ate prompt will appear on boot-up.

2. Linker: When preparing a program which requires repeated compilation and includes a file to be linked, this is a useful aid. Once this program has been used, the "EXEC/AUTOLINK" can be directly executed as often as required during the editing session.

3. Hardcopy: This program will transfer multiple files to "CONSOLE:" or "PRINTER:" using the direct F(iler - T(ransfer option. This is an example of text file transfer (to the printer) without having to directly OPEN the printer file (i.e.-REWRITE(FID, 'PRINTER:'). Following each transfer, a manual form feed is suggested. This has not been included to obviate having to enter and leave the F(iler repeatedly.

```
   (*************************************)

PROGRAM BOOTDATE;
USES CHAINSTUFF;
VAR XFILE : TEXT;
BEGIN
   REWRITE (XFILE, 'DATE.TEXT');
   WRITE (XFILE, '%FD%%');
   CLOSE (XFILE, LOCK);
   SETCHAIN ('EXEC/DATE')
END.

(*************************************)

(*  Generates an EXEC file which acts as an
automatic linker *) (*  for  Regular  units
and  External PROC/FUNCs.This can be use-*)
(* ful where repeated editing of a  program
requiring linkage*) (* is necessary.Version
2: 2/84 *)
PROGRAM XLINKER;
USES CHAINSTUFF;
VAR
   XFILE : TEXT;
BEGIN
         (* Open an EXEC file
            named AUTOLINK *)
   REWRITE (XFILE ,'AUTOLINK.TEXT');
         (* Invokes Linker *)
   WRITE (XFILE,'%L');
         (* Prompt for Host File *)
   (* Assumes host file is
      System.Wrk.Code *)
      (* Else enter WRITELN (XFILE,
      'FILENAME.CODE');
   WRITELN (XFILE);
   (* Prompt for Lib File *)
   WRITELN (XFILE, '#5:NAME OF FILE
            TO BE LINKED');
   (* Next two Prompts:Lib File?
      & Map File? usually <RET>s *)
   WRITELN (XFILE);
   WRITELN (XFILE);
         (* Prompt for Output File? *)
   (* If Host File is Sys.Wrk.Code
    then next line = WRITELN(XFILE); *)
   (* Else enter name of output file-
    WRITELN (XFILE,'FILENAME.CODE')*);
   WRITELN (XFILE,'%%');
   CLOSE (XFILE,LOCK);
   SETCHAIN ('EXEC/AUTOLINK')
END.
(*************************************)

(* Using EXEC FILE to transfer multiple
 text files to printer *)
PROGRAM HARDCOPIES;
USES CHAINSTUFF;

CONST MAX = 5;  (* Max number of files
                   to be transferred *)

VAR XFILE : TEXT; NUMBER, I : INTEGER;
            FILENAME : STRING;
    STORE : ARRAY[1..MAX OF STRING;

PROCEDURE GETALLFILENAMES;
VAR CH : CHAR; CHTOSTR : STRING[1;
         I : INTEGER;
BEGIN
   PAGE (OUTPUT);
   WRITE ('How many Files to print?  ');
   READ (KEYBOARD, CH); WRITELN;
   CHTOSTR[1 := CH;  (* Convert char
                        to string *)
   (* Convert string to number to
      avoid input error *)
   NUMBER :=  NUMBER * 10 +
           ORD (CHTOSTR[1) - ORD ('0');
   IF (NUMBER < 1) OR (NUMBER > MAX) THEN
      BEGIN
         WRITELN ('Inaccurate number
                  entry...Exiting program!');
         EXIT (PROGRAM)
      END;
   FOR I := 1 TO NUMBER DO
      BEGIN
         WRITELN ('File #',I,': ');
         WRITELN ('Enter complete
                   Filename');
         READLN (FILENAME);
         IF POS ('.TEXT', FILENAME) = 0
                        THEN
            FILENAME := CONCAT (FILENAME,
                                '.TEXT');
```

```
        STORE[I := FILENAME
      END
END; (* GETALLFILENAMES *)

PROCEDURE MAKEXECFILE;
BEGIN
   REWRITE (XFILE, 'FILETRANS.TEXT');
   WRITE (XFILE, '%F');
   FOR I := 1 TO NUMBER DO
      WRITELN (XFILE, 'T', STORE[I,',
               CONSOLE:');(* OR PRINTER:*)
   (* Remove Exec file when completed *)
   WRITELN (XFILE, 'R','FILETRANS.TEXT');
   WRITELN (XFILE, 'YQ%%');
   CLOSE (XFILE, LOCK)
END; (* MAKEXECFILE *)

BEGIN
   NUMBER := 0;
   GETALLFILENAMES;
   MAKEXECFILE;
   SETCHAIN ('EXEC/FILETRANS')
END.

C/FILETRANS')
END.
```



Abbreviated Keys For Random Access

-Max Nareff.

This short program illustrates one method of using shortened keys and the wildcard symbol ("=") to access elements of an array (i.e. - from an address file, etc.).

For demo purposes the array will be kept simple; a short array of records constituting a bare - bones address file. The user is requested to enter the E(ditor and prepare 5 records, each consisting of 4 fields. Each field is typed on one line without any intervening blank lines. The consecutive fields are a (capitalized) Name (JAMES B. BONES), street address, city,state and zip, and lastly, a phone number. The file is to be saved as "ADDBOOK" on drive #5.

When an access Key is input, it is first capitalized, if necessary, and its length determined. In a standard loop the Key is then compared with a corresponding length of the Name field. Concordance results in the screen appearance of the entire selected record.

PROCEDURE DIRECTOR directs any necessary capitalization of the Key and also determines whether the wildcard ("=") will be operative. The important selecting statement is in PROCEDURE FINDNSHOW. If S, the Key, is lexicologically equal to the same length of the Name then write the record (IF S := COPY (NAME, 1, LENGTH(S) THEN WRITE RECORD).

This then demonstrates how a simple address file can be prepared and used with abbreviated keys.

```
(* Random access using shortened
 Keys or the wildcard symbol("=") *)
(* Program by Max J. Nareff 3/84 *)

PROGRAM SHORTKEY;
CONST
   ARRAYSIZE = 5;  (* Change to suit *)
TYPE
   SIZE = 1..ARRAYSIZE;
   DIARY = RECORD
              NAME : STRING;
              ADDR : STRING;
              CITY : STRING;
              PHON : STRING
            END;
   INFO = PACKED ARRAY[SIZE OF DIARY;
VAR
   BOOK : INFO;
   KEY : STRING;
   LEN : INTEGER;

   (* Capitalizer *)
FUNCTION CAPS (CH:CHAR):CHAR;
BEGIN
   IF ('a' <= CH) AND (CH <= 'z') THEN
      CAPS := CHR (ORD(CH) - 32)
   ELSE
      CAPS := CH
END; (* CAPS *)

   (* Searches & Displays *)
PROCEDURE FINDNSHOW (VAR S:STRING);
VAR
   I : INTEGER;
   CH : CHAR;
BEGIN
   FOR I := 1 TO ARRAYSIZE DO
      WITH BOOK[I DO
         BEGIN
(* If capitalized Key compatible with
      equal length of Name then *)
         IF S = COPY (NAME, 1, LEN) THEN
          BEGIN
            WRITELN (NAME);
            WRITELN (ADDR);
             WRITELN (CITY);
             WRITELN (PHON);
             WRITELN;
             REPEAT
              WRITE ('Search continued
                 -> (Y/N)? :   ':45);
               READ (KEYBOARD, CH);
                    WRITELN; WRITELN
              UNTIL CH IN ['Y', 'y',
                      'n', 'N'];
              IF CH IN ['N', 'n' THEN
                    EXIT (PROGRAM)
          END
      END;
 WRITELN; WRITELN ('Search complete!':30)
END; (* FINDNSHOW *)
```

```
PROCEDURE WRITEALL;
VAR
   I : INTEGER;
BEGIN
   FOR I := 1 TO ARRAYSIZE DO
      WITH BOOK[I DO
         BEGIN
            WRITELN (NAME);
            WRITELN (ADDR);
            WRITELN (CITY);
            WRITELN (PHON)
         END
END; (* WRITEALL *)

PROCEDURE READARRAY;
VAR
   I : INTEGER;
   STR : STRING;
   FID : TEXT;
BEGIN
   (*$I- *)
   RESET (FID, '#5:ADDBOOK.TEXT');
   (*$I+ *)
   IF IORESULT <> 0 THEN
      BEGIN
         WRITELN;WRITELN ('Exiting program
           Unable to locate Address File.');
         EXIT (PROGRAM)
      END;
   FOR I := 1 TO ARRAYSIZE DO
      WITH BOOK[I DO
         BEGIN
            READLN (FID, STR);
            NAME := STR;
            READLN (FID, STR);
            ADDR := STR;
            READLN (FID, STR);
            CITY := STR;
            READLN (FID, STR);
            PHON := STR
         END;
   CLOSE (FID)
END; (* READARRAY *)

(* Selects option for Random Access *)
PROCEDURE DIRECTOR;
VAR
   I : INTEGER;
BEGIN
   IF LEN < 1 THEN EXIT (PROGRAM);
   (* If needed capitalize Key entry *)
   FOR I := 1 TO LEN DO
      KEY[I := CAPS (KEY[I);
   IF (POS('=', KEY) = 1) THEN WRITEALL;
               (* Wildcard option only *)
   IF (LEN >= 1) AND (POS('=', KEY) = 0)
                     THEN
      FINDNSHOW (KEY)
   ELSE
      (* Handles shortkey + "=" or just
            shortkey alone *)
      IF (POS('=', KEY) > 1) THEN
         BEGIN
            DELETE (KEY, POS('=', KEY), 1);
            LEN := LEN -1;
            FINDNSHOW (KEY)
         END
END; (* DIRECTOR *)

BEGIN (* MAIN *)
   PAGE (OUTPUT);
   WRITELN; WRITELN
('This is a Random Access Demo using
 shortened keys or the wildcard symbol');
   WRITELN; WRITELN ('Attempting to read
                      Disk Info.'); WRITELN;
   READARRAY;
   WRITE ('<CR> to Abort, or Enter a short
                   Key, or "=" for all:  ');
   READLN (KEY);
   LEN := LENGTH (KEY);
   WRITELN;
   DIRECTOR
END.
```

LISTEN to
"The Voice of Experience"

SynPhonix™

Electronic Speech Articulator for the Apple!

Hear at last, hardware synthesis at software prices. SynPhonix™ produces unparalleled voice quality using the ***all new SSI263*** phoneme synthesizer with 64 phonemes, 100's of allophones, plus software controlled pitch, amplitude, rate and filter. The SONIX™ 65© Speech Operating System puts you in control of it all. SynPhonix™ is brought to you by professionals with more than a decade of experience in speech synthesis products. Applications are abundant. SynPhonix™ features include:

- SSI263 Phoneme Synthesizer
- Unlimited vocabulary
- High Quality Phonetic Voice
- Low Data Rate
- Singing/Sound Effects
- Comprehensive User's Manual
- Demo Software Included
- Free Spelling Program
- Transparent to User's Software
- SONIX™ 65©
  - Speech Operating System
  - Speech Editor
  - Preprogrammed Vocabulary
  - Vocabulary File Manager
  - EASY TO USE
- 90 Day Limited Warranty
- Suggested Retail $125.00

Artic Technologies
"The Voice of Experience"

Dealer inquiries welcome!

2234 Star Ct.-2D (313) 852-8344
Auburn Heights, Michigan 48057

Apple is a Trademark of Apple Computer, Inc.

The Graphic Solution

Jerry Matsukado, Copyright 1984

THE GRAPHICS SOLUTION (hereinafter referred to as TGS) by Accent Software (4546 El Camino Real, Suite S, Los Altos, Ca 94022) is a graphics editor and animation system. Conceptually, it works like this:

"Shapes" are created (drawn and edited). "Pictures" of these shapes -- called "frames" -- are taken; and a series of these frames are stored as "sequences". In short, it's just like making a movie.

Originally, the 48K Apple II version was being reviewed -- that version relies on keyboard input for image creation and program control. However, Accent made available the new 64K version, which permits use of an expansion module (available at extra cost), allowing you to create images using the Koala Pad. This version is the subject of this discussion. There is also an expansion module available supporting use of the Gibson Light Pen.

What it does

The system is designed for development of quality graphics and animation for presentation purposes in such areas as education, sales, product demonstration, marketing, artwork design, etc.

You draw/create shapes using either the keyboard or one of the optional expansion modules. You can edit these shapes; get them to move in various ways (expand, contract, move along various paths, etc.); change color.

You can project these shapes and their animated paths onto various backgrounds. Animation speeds can be varied, especially effective in developing dynamic flowcharting, and 3-D projections. These are illustrated on the TGS demonstration disks (included).

Documentation

First, recognize that, while you can use The Graphics System (TGS) to draw pictures, do some animation, etc., more or less by rote, to utilize its capabilities properly and effectively is going to require some work. To this end, the documentation will be of great assistance.

The documentation is straightforward and well thought-out. And while it can be a bit unwieldly from time to time, that is probably the nature of the subject matter -- after all, graphics is not necessarily an easy subject.

The system and its operating environment are clearly and logically presented, so that newcomers to graphics quickly grasp key concepts. Terms are generally defined as you go along; thus minimizing "look-ahead" searches for buzzwords used but not yet defined.

The entire manual is written in a tutorial fashion, and, to be effective, you really ought to have the system booted as you step through the various sections. It is comforting to know that, within about the first forty pages, most of the key concepts will have been presented, and the user been made to utilize them. Actually, the tutorials encompass the first 130 or so pages, by which time all of the capabilities of the system are presented. There's lots to learn.

For those of us who forget a key definition from time to time, a glossary would be helpful -- or at least an index. A one-concept, one-page style is used, so the table of contents can be scanned to locate key terminology. However, something in alpha-sequence would definitely be helpful.

As each section/chapter is completed, there is a summary of the concepts presented therein; and a series of exercises. This is an excellent way to learn and reinforce concepts just presented; and -- don't panic -- where answers are required, Appendix B lets you know if you're right or not.

Overall, the documentation is well-presented; logically laid-out; well-written; and remarkably free of spelling and syntax errors (three typos were detected). Most importantly, it is effective and should bring you up to speed in short order.

Presentation

TGS comes on two diskettes, each containing a one-sided TGS Program disk. You, therefore, have a backup disk immediately available. On the reverse of one program disk is a demo; on the reverse of the other is a sequence disk which presents tutorials of some system features.

Use of the Demo and the Sequences disks show, graphically, what the system can do, and provide the command sequences which produced them. Do not underestimate the value of these demos and sequences.

The Sequence disk, for example, gives a method by which you can create shades of colors beyond those directly available from TGS. Another example is the way the Sequence disk illustrates the creation of 3-D images using TGS features.

The Expansion Module for either the Koala Pad or the Gibson Light Pen comes on a separate disk and is available at extra cost.

System Structure

You get to the various TGS modes from the menu. Broadly speaking, the menu provides options which allow you to create and maintain sequences; access backgrounds; create circles; do file maintenance (e.g., catalog, file deletion, etc.); run the projector module (used for displaying animations under your control); and access expansion modules. Images may be drawn in either hi- or low-res modes.

You draw pictures using the Koala Pad or other supported input device in hi-res mode; then do your detail editing in low-res (magnified mode). Low-res mode effectively "magnifies" a subsection of the full-screen hi-res image; thus allowing detail editing to be done. The resultant edited image is then projected back on the hi-res screen.

Graphics output are stored on disk in DOS 3.3 format. Both sequences and backgrounds may be stored.

System Capabilities

The System has been developed in a manner analogous to making a film: Create a variety of shapes; then take a series of "pictures" (termed "frames") of these shapes. Put them together in a sequence, and you have animation. Superimpose those sequences on a background, and you have a film.

In actual usage, the end result is a hi-res screen in which certain parts may move; text may appear; colors may change; objects may expand or contract -- any number of things.

The entire screen is used in hi-res mode. TGS allows for a window to be superimposed on the hi-res screen, capturing a segment of the hi-res image. Think of the window as representing a piece of film, and think of the hi-res screen as a landscape.

When you look through the viewfinder of a camera at the landscape, you only see a portion of the landscape; so that when the film is exposed, only a portion of the total image is photographed.

In like fashion, TGS creates shapes by placing the hi-res window (the "film") over a portion of the hi-res screen (the "landscape"). A shape is created by "taking a picture" of what is in the hi-res window.

The shape (i.e., whatever is in the hi-res window) can be edited by going to low-res mode. There, the shape is expanded to fill the screen, and, through use of the keyboard, detail changes can be made. If you have the expansion module, detail changes can be made using, for instance, the Koala pad.

As shapes are created, "pictures" are taken of them, and these "pictures" are stored in tables. They can then be played back, and timings can be altered so that images are projected on the screen for varying lengths of time. The result of this activity is the creation of animated sequences.

Selected System Features

The Koala Expansion Module greatly facilitates the process of graphics creation. The Koala Pad substitutes for the keyboard, and makes drawing images much easier. Modes for line-creation and drawing are supported; and a series of eight colors in two sets of four are provided for.

Macro Facility: Animation sequences may involve entry of repeated sets of commands. Besides being boring to do, the more commands that must be entered, the greater the chance of error. TGS allows you to develop a series of commands and store them in a macro.

Then, when you want to create a certain sequence, you execute the macro once to execute all of the commands contained in the macro, rather than having to laboriously code and recode the command set.

Text: Text may be easily created through the keyboard at any time. From the hi-res mode, press Control-T, and you are in text mode. At that point, start typing, and you have text. You can place text anywhere in the hi-res screen that you wish.

Place the hi-res window where you want it to start; and use standard commands to start new lines, etc. Text, under TGS, can be treated as any other shape -- which means you can change its color, animate it, even move it around at varying speeds.

Help: Until you become familiar with the way TGS operates, there are going to be times when you'll be doing something and want to do something else. Hitting ESC(ape) in any mode, prompts a help sequence which lists all the appropriate commands for getting out of whatever you're doing, giving you a series of descriptive prompts which get you where you want to go.

Speed Modification: TGS allows you to vary the timing at which shapes are projected, allowing you to interject pauses in your sequences, hold an image for a pre-determined period of time before changing its size, color, and/or shape.

Multiple Shape Animation: You can do multiple-shape animation. However, you will have to keep track of the various shapes; and that poses a whole new set of problems. The documentation does provide guidance in handling these problems, though; so if this is something you want to do, TGS will help you do it.

Projector: This permits an Applesoft program to load sequences and display them. Samples of Applesoft programs which would invoke the Projector are provided, together with very comprehensive explanations on a line-by-line basis.

DOS 3.3 Format Storage: TGS stores backgrounds and sequences in standard DOS 3.3 format. This does result in increased storage usage since the entire screen image must be stored (for background images). However, because you can use backgrounds created by other systems, you don't have to start from scratch, recreating graphics you may have created earlier.

Examples of such may be graphics created with expanded color sets (as mentioned, the Koala Expansion Module gives you a maximum of eight colors); they may be really outstanding examples of computer graphics (yours, of course); and/or, possibly, a pain to redo. If they're in 3.3 format, load them up and go.

Variable Hi-Res Windows: This is a very powerful facility within TGS. The example given is having a city map as a background, with small shapes representing cars travelling on its streets. Use of variable hi-res windows allows this to be done without having to capture the background (the city map) as part of the shapes. This saves a lot of disk space.

Summation

TGS has a lot of features which permit the development of quality graphics, and their presentation in a professional manner.

For graphics tyros, it offers a lot of features which will enable them to achieve their goals.

For those new to graphics, I believe that the quality of the documentation, inclusive of the Demo and Sequences disks, provides an excellent opportunity to become familiar with graphics concepts while working on specific projects, or while just playing around.

The Koala Expansion Module smooths out the mechanical process of image-creation, and makes it a lot more fun to do.

The only real criticism to be offered is in the area of I/O error diagnostics. For example, if you get a real I/O error, or -- less seriously -- if you attempt to load a sequence that's not there, in either case the screen simply flashes the word "ERROR" a few times, then returns you to the menu. Not too helpful.

I understand that the reason for this is that there's so much program comprising TGS that there's not a whole lot of disk space left for error diagnostic routines. But from what I've seen of TGS, it's probably only a matter of time until the folks at ACCENT take care of that little glitch. Until they do, as in the case of loading the sequence that's not there, I'll just have to be more careful.

Poke of the Month

Kathy Rudovsky

POKE 34,N (N representing a number between 0 and 23) will set the top margin of the text window. Why do this??? First of all because your computer is capable of doing it. If you would like a title on the screen that stays while all else scrolls under it.....that's how!!!

Interest Group News

Macintosh Group - The Macintosh Group of the San Francisco Apple Core has expanded to the East Bay and will hold its first meeting Thursday, August 23 at 7:00.

The meeting will be held in the meeting room at **Breuner's** Furniture Store, 3250 Buskirk Ave., Pleasant Hill.

For more information regarding this group contact Stan Guidero at 432-9713.

INTEREST GROUP NEWS

The East Bay Group is still in search of a new location. If anyone has any suggestions or recommendations please contact Lenny Leff and/or Kathy Rudovsky.

Green Apples, the BASIC language course, is going to start a new series sometime in September. Interested members should contact Keith Thomas for more information.

Graphics is in the process of changing its meeting location to the clubhouse.

Nu'ers (New Users), an interest group dedicated to assisting members in all subjects of the APPLE in an atmosphere that is informal and friendly.

MEMBERS!!!!! BE ADVISED TO CHECK WITH INTEREST GROUP LEADERS OR THE SFAC CLUBHOUSE FOR THE MOST RECENTLY UPDATED INFORMATION REGARDING INTEREST GROUP MEETINGS.

INTRO TO THE APPLE COMPUTER

Thanks to the expertise of the seminar instructors, the introductory mini-seminars were very successful.

Many thanks to Michael Fischer, LOGO; Stan Dinsmore, Assembly Language; Keith Thomas, Applesoft Basic; R. Dudley Ackerman, FORTH; Michael Schnapp, PASCUAL; Reed Harris, Graphics; and Patrick Taffe (Computer Connection), Word Processing and Business Applications.

Thanks too, to Ron Dell'Aquila who came to the rescue not once, but twice during the seminars with the proper hardware necessary to carry on.

Telecommunications had to be cancelled at the last minute, but is rescheduled for Monday September 17th.

Two new seminars in the works for September or October will be "PEEKS POKES & CALLS - WHAT ARE THEY?" and "WPL" (Word Processing Language with Apple Writer][). Pre-registration will be required. More details will be available at the General Meeting.

For Sale

MicroModem //e - 4 months old.
Includes SmartCom I and other software.
Asking $200.00.
Call 564-0848, after 7:00.

The San Francisco Apple Core

Presents

The 4th Annual Apple Swap Meet

September 15th

Members Free
Non-Members Admission $1.00

Building C - Fort Mason
San Francisco, CA
Room 100, 10:00 am to 2:00 pm

Sales end at 1:00 pm and Dealer Auction starts at 1:15 pm

Booth space available to individual members	$25.00
Booth space available to dealers	50.00

For sales information and booth space contact:
Don Scellato (415) 272-8895 or (415) 672-2051 eve
Ken Silverman (408) 988-5594 days

A booth is 8' x 10'

Hundreds of NEW & USED Software & Hardware Items

Consignment table available to members who have only a few items to sell - Contact Don Scellato (415) 672-2051 eve's.

Thanks from Max

Dear Fellow Club Members,

Thank you forthe Bronze Apple. The Club's appreciation bears a special meaning for me.

From my point of view, it is I who appreciates what being a participating club member really means - a great deal of satisfaction, and the unique opprtunity inspired by the club, at my age, to learn a new and exciting method of communication and education.

May I express my deep appreciation for the untiring efforts of all of you and the other selfless volunteers, who really bear the burdens of managing and operating a genuine local community resource.

Again, many thanks.

Max Nareff.

SAN FRANCISCO APPLE CORE
ANNUAL REPORT

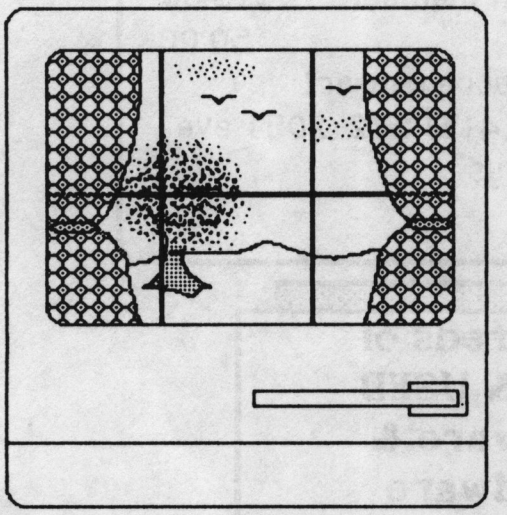
To the Membership
San Francisco Apple Core

The following is a summary of the financial activity of the San Francisco Apple Core for the year ended March 31, 1984.

Revenues	$	41,965
Costs and Expenses:		
Cider Press		13,139
Merchandise for Resale		11,038
Office Staff		5,632
Rent - Office & Ft. Mason		4,485
Fairs		2,501
Postage		2,353
Other expenses		2,872
Total Expenses		42,020
Excess of Expenses over Revenues	$	55

Respectfully submitted,

Alan Morris, Treasurer



Windowing on the Mac.

Lisa	Larry Clopp 333-3707	Call for location	3rd Thursday	7:30 pm
Marin CP/M	Jim Ayres 383-0473 383-5759	301 Poplar Mill Valley	2nd & 4th Tuesday	7:30 pm
S.F. CP/M	Ron Dell'Aquila 566-8460	Clubhouse	4th Wednesday	7:30 pm
Dot Matrix Printers	Michael Fischer (w) 557-1250 (h) 933-4435	Clubhouse	4th Tuesday	7:30 pm
Forth	R. Dudley Ackerman 626-6295	Clubhouse	1st & 3rd Tuesday	7:30 pm
Pascal	Mike Schnapp 386-2238	Call for location	4th Monday	7:30 pm
Nuer's	Kathy Rudovsky 756-6480	Clubhouse	3rd Wednesday	7:30 pm
Spreadsheets	Don Scellato 672-2051	Clubhouse	2nd Tuesday	7:30 pm
East Bay Spreadsheets	Don Scellato 672-2051	Bruener's - Buskirk Pleasant Hill	4th Monday	7:00 pm
Green Apples	Keith Thomas 837-3164	Clubhouse	1st 4th & 5th Thursday	8:00 pm
East Bay	Lenny Leff 526-9541	Call for location	1st Tuesday	7:30 pm
MacIntosh	Don Scellato Alan Smerican 672-2051	Fort Mason Bldg. C	3rd Saturday	12:00 pm
Product Review	Angela Allberry (w) 654-9351 (h) 391-2715	Clubhouse	2nd Thursday	7:30 pm
Telecommunications	George Thornally 398-7926 776-2603	Call for location	2nd Tuesday	7:30 pm
Graphics	Elliot Isenberg 282-9733	Call for location	2nd Monday	7:30 pm

San Francisco Apple Core Clubhouse
1515 Sloat Blvd. Suite 2
San Francisco, CA 94132
(415) 566-2342

SF APPLE CORE
1515 Sloat Blvd., Suite 2
San Francisco, CA. 94132

TIME VALUE

Bulk Rate
U.S. Postage
PAID
Permit #85
El Cerrito, CA 94530

TO:

3121AUG84
DOUGLAS STORKOVICH
921 PINTO CT
WALNUT CREEK CA 94596